

## WHAT IS CLAIMED IS:

1. A fault monitoring method in a computer including a first operating system (OS), a second OS, and a multi-OS controller having a communication function between said first OS and said second OS, said fault monitoring method comprising the steps of:

loading said first OS, said second OS and said multi-OS controller into a memory included in the computer;

executing said first OS, said second OS and said multi-OS controller by using a processor included in the computer under control of said multi-OS controller;

successively receiving, in a fault monitor activated by execution of a program on said second OS, an alive message indicating an alive state of said first OS transmitted from said first OS by using the multi-OS controller; and

restarting said first OS, when a reception interval of said alive message has exceeded a predetermined time in said fault monitor.

2. A fault monitoring method according to claim 1, wherein said fault monitor determines whether said alive message has exceeded said predetermined time by comparing the reception interval of said alive message with predetermined time information stored in said memory.

3. A fault monitoring method according to claim 2, wherein said fault monitor restarts said first OS in accordance with a restart method stored in said memory by using said multi-OS controller loaded in said memory.

4. A fault monitoring method according to claim 1, wherein said multi-OS

controller manages said resources by dividing said resources to provide divided resources to said first OS and said second OS, respectively.

5. A fault monitoring method according to claim 1, wherein said communication function between said first OS and said second OS includes a function in which one OS calls a process to be executed by the other OS.

6. A multi-operating system (OS) computer having a fault monitoring function, comprising:

a processor;

a memory;

a first OS loaded on said memory by said processor;

a second OS loaded on said memory by said processor;

a multi-OS controller, having a inter-OS communication function between said first OS and said second OS, loaded on said memory by said processor; and

a fault monitor activated by execution of a program on said second OS,

wherein by using said inter-OS communication function of said multi-OS controller, said fault monitor successively receives an alive message indicating an alive state of said first OS transmitted from said first OS, and when a reception interval of said alive message has exceeded a predetermined time, said fault monitor restarts said first OS.

7. A multi-OS computer according to claim 6, wherein said fault monitor determines whether said alive message has exceeded said predetermined time by comparing the reception interval of said alive message with predetermined time

information stored in a memory included in said computer.

8. A multi-OS computer according to claim 7, wherein said fault monitor restarts said first OS in accordance with a restart method stored in said memory by using said multi-OS controller loaded in said memory.

9. A multi-OS computer according to claim 6, wherein said multi-OS controller manages resources by dividing said resources to provide divided resources to said first OS and said second OS, respectively.

10. A multi-OS computer according to claim 6, wherein said communication function between said first OS and said second OS includes a function in which one OS calls a process to be executed by the other OS.

11. A computer readable storage medium for storing a program, said program comprising:

a first operating system (OS);

a second OS different from said first OS;

a multi-OS control program unit for managing computer resources, said multi-OS control program unit having an inter-OS communication function between said first OS and said second OS; and

a fault monitoring program unit for monitoring a fault of said first OS via said inter-OS communication function, said fault monitoring program unit operating on said second OS,

wherein said first OS, said second OS, said multi-OS control program unit and

said fault monitoring program unit are loaded in a memory included in said computer.

12. A computer readable storage medium according to claim 11, wherein a multi-OS controller executes said multi-OS control program unit to manage said resources by dividing said resources to provide divided resources to said first OS and said second OS, respectively.

13. A computer readable storage medium according to claim 11, wherein said inter-OS communication function includes a function in which one OS calls a process to be executed by the other OS.

14. An operating system (OS) fault monitoring method for a computer, said computer including a first OS, a second OS different from said first OS, a multi-OS controller for managing computer resources, said multi-OS control unit having an inter-OS communication unit between said first OS and said second OS, and a fault monitor operating on said second OS, said OS fault monitoring method comprising the steps of:

loading said first OS, said second OS, said multi-OS controller and said fault monitor in a memory included in said computer;

transmitting an alive message from said first OS to said fault monitor via said inter-OS communication unit and said second OS; and

determining whether said alive message has been received by said fault monitor within a predetermined time.

15. An OS fault monitoring method according to claim 14, wherein when

said alive message is not received by said fault monitor though a predetermined time has elapsed, said first OS is restarted in accordance with a preset OS recovery method stored in said memory by using said multi-OS controller and said fault monitor.

16. An OS fault monitoring method according to claim 14, wherein said multi-OS controller manages said resources by dividing said resources to provide divided resources to said first OS and said second OS, respectively.

17. An OS fault monitoring method according to claim 14, wherein said inter-OS communication function includes a function in which one OS calls a process to be executed by the other OS.